**CSS**

层叠样式表(英文全称：Cascading Style Sheets)是一种用来表现HTML或XML等文件样式的计算机语言。CSS不仅可以静态地修饰网页，还可以配合各种脚本语言动态地对网页各元素进行格式化。

**CSS选择器**

selector {property: value}

选择器指明了样式的作用对象，用于匹配HTML元素

**选择器分类**

元素选择器 a{}

伪元素选择器 ::before{}

类选择器 .link{}

属性选择器 [type=radio]{}

伪类选择器 :hover{}

ID选择器 #id{}

组合选择器 [type=checkbox]+label{}

否定选择器 :not(.link){}

通用选择器 \*{}

**伪元素和伪类的区别**

伪元素 ：：before ：：after

伪元素在DOM树中创建了一些抽象元素，这些抽象元素是不存在于文档语言里的。伪元素是真实存在的，可以有内容和样式。

伪类 ：hover ：focus

伪类存在的意义是为了通过选择器找到那些不存在与DOM树中的信息以及不能被常规CSS选择器获取到的信息。伪类相当于选中的一种状态。

**选择器权重**

**选择器本身的权重 加法不进位**

ID选择器 #id{} +100

类 属性 伪类 +10

元素 伪元素 +1

其他选择器 +0

**其他情况**

！important 优先级最高

元素属性 优先级高 内联样式

相同权重，后写的生效

**非布局样式**

字体 font-family

自定义字体的使用场景：品牌+图标

行高 line-height

背景 background

雪碧图，图标放在一张图上，设定大小、位置

雪碧图作用：减少HTTP请求数，提高加载性能

边框 border

三角形利用边框可以做，左右边框透明，底边长很小就可以，扇形可以使用边框圆角

滚动

overflow –>auto自动 hidden滚动条隐藏 scroll滚动条显示 visible内容全显示

文字折行

overflow-wrap（word-wrap）通用换行控制

word-break针对多字节文字

white-sapce 空格处是否断行

装饰性属性及其他

字重（粗体）font-weigth

斜体 font-style：itatic

下划线 test-decoration

指针 cursor

图片编码base64

减少请求、适用于小图片、体积变大1/3

**CSS Hack**

Hack即不合法但生效的写法

主要用于区分不同的浏览器 IE

缺点：难理解、难维护、易失效

替代方案：特性检测、针对性血加class

**checkbox美化**

input：checked+label

利用背景，背景设置为图片，大小、位置。

隐藏原生input checkbox：style="display:none”

**CSS布局**

早期table

后来以技巧性布局为主

现在有flexbox/grid

响应式布局是必备知识（适配）

**常用布局**

table表格布局

float浮动+margin 兼容性好

inline-block布局

flexbox布局

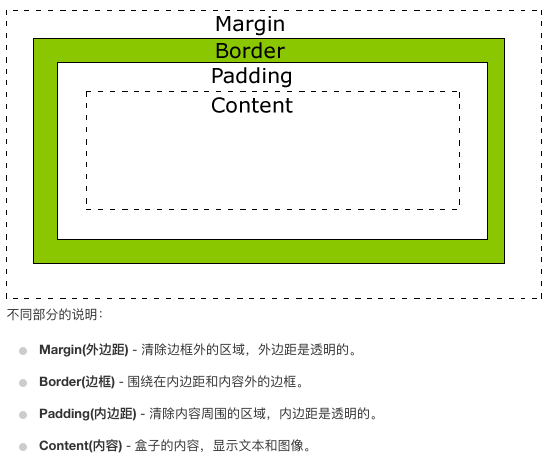
**盒模型**

Margin(外边距) - 清除边框外的区域，外边距是透明的。

Border(边框) - 围绕在内边距和内容外的边框。

Padding(内边距) - 清除内容周围的区域，内边距是透明的。

Content(内容) - 盒子的内容，显示文本和图像。



**display/position**

确定元素的显示类型

block/inline/inline-block

确定元素位置

static静态/relative相对/absolute绝对/fixed固定

static：默认情况，按照文件流布局

relative：相对元素本身的偏移，不改变原来占据的空间

absolute ：脱离文件流，独立存在，就近原则，相对于最近的relative、absolute

fixed：从文件流脱离，不影响其他，独立存在，相对于可视区域viewport

z-index属性

层叠顺序relative 、absolute、fixed

**flexbox**

弹性盒子

盒子本来就是并列的

指定宽度即可

使用：选择器中display设置为flex，然后在对flex进行设置

**float**

元素“浮动”

脱离文档流

但不脱离文本流（排挤文字）

对自身的影响：形成块，位置尽量靠上、靠左(右)

形成块：span本身没有行高，设置了float后可以设置行高

对兄弟的影响：上面贴非float，旁边贴float，不影响其他块级元素位置，影响其他块级元素内部文本

对父级元素的影响：从布局上“消失”，高度塌陷

使用：选择器中float:left

**清除浮动**

让盒子负责自己的布局

overflow:hidden(auto)

::after{clear:both}

**inline-block**

像文本一样排block元素

需要处理间隙问题

父元素font-size:0 inline-block中字体重新设置

自适应上面不太方便

**响应式设计和布局**

在不同设备上正常使用

一般主要处理屏幕大小问题

主要方法：隐藏+折行+自适应空间

rem/viewport/media query媒体查询

viewport

<meta name="viewport" content="width=device-width, initial-scale=1.0">

media query

@media (max-width: 640px) {

.left{

display: none;

}

}

**如何适配移动端页面**

viewport

rem/viewport/media query媒体查询

设计上：隐藏+折行+自适应空间

**效果属性**

box-shadow

text-shadow 文本投影

border-redius边框圆角

background

clip-path

**box-shadow**

**参数**

**inset** 有inset 则为内阴影，没有insert 则为外阴影，默认为外阴影。

**offset-x** 横向阴影的大小，正值阴影在右边，负值阴影在左边，0的时候阴影中盒子后面，看不见的，如果有blur-radius值会有模糊效果。

**offset-y** 纵向阴影的大小，值同offset-x。

**blur-radius** 阴影的模糊程度，值越大，阴影越模糊。

**spread-radius** 阴影的扩大缩小，正值时，阴影扩大；负值时，阴影缩小，默认为0，和盒子同样大。

**目的**

营造层次感

充当没有宽度的边框

特殊效果

**border-redius**

圆角矩形

圆形 border-redius:50%

半圆/扇形 控制一个角或两个角

奇怪的角

**background**

纹理、图案

渐变 background: linear-gradient();

雪碧图动画

动画效果，放在同一张图上，位置移动，速度设置 transition hover

背景图尺寸适应

**clip-path**

对容器进行裁剪

常见几何图形

自定义路径

**3D变换**

2D变换：translate位移，scale缩放，skew斜切，rotate旋转

3D变换：在3D中进行变换

**如何用一个div画XXX**

box-shadow无线投影

::before

::after

**如何产生不占空间的边框**

box-shadow

outline

**如何实现IOS图标的圆角**

clip-path: (svg)

**如何平移/放大一个元素**

transform：translateX(100px)

transform：scale(2)

**如何实现3D效果**

perspective：500px

transform-style：preserve-3d

transform：translate rotate

**CSS中的动画类型**

transition补间动画

keyframe关键帧动画

逐帧动画

**补间动画**

位置-平移

方位-旋转transform

大小-缩放transform

透明度opacity

其他-线性变换transform

**关键帧动画**

animation keyframe

相当于多个补间动画

与元素状态的变化无关

定义更加灵活

**逐帧动画**

特殊的关键帧动画

适用于无法补间计算的动画

资源比较大

使用step(1) 去掉补间动画，动画时长比较小

**过渡动画和关键帧动画的区别**

过渡动画需要状态变化

关键帧动画不需要状态变化

关键帧动画能控制更精细

**如何实现逐帧动画**

使用关键帧动画

去掉补间 animation-timing-function：steps(1)

**CSS动画的性能**

性能不坏

部分情况下优于JS

但JS可以做到更好

部分高危属性 box-shadow等

**CSS预处理器**

基于CSS的另一种语言

通过工具编译成CSS

添加了很多CSS不具备的特性

能提升CSS文件的组织

less sass

**预处理器的能力**

嵌套 反映层级和约束

变量和计算 减少重复代码

Extend和Mixin代码片段

循环 适用于复杂有规律的样式

import CSS文件模块化

**常见CSS预处理器**

Less（Node.js）

Sass（Ruby 有Node版本）

**预处理器作用**

帮助更好地组织CSS代码

提高代码的复用率

提升可维护性

**预处理器优缺点**

优点：提高代码的复用率；提升可维护性

缺点：需要引入编译过程；有学习成本